BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-334055
(43)Date of publication of application : 18.12.1998

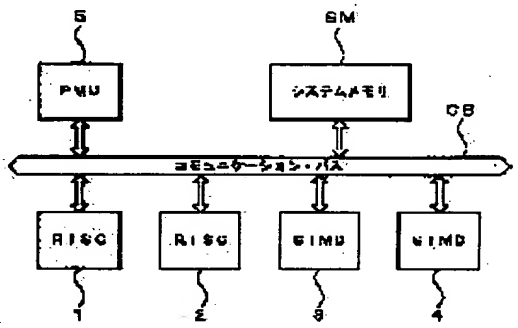
(51)Int.Cl. G06F 15/16

(21)Application number : 09-145383
(22)Date of filing : 03.06.1997

(71)Applicant : SONY CORP
(72)Inventor : TAMURA YUKIHIRO

(54) MULTIPROCESSOR SYSTEM

(57)Abstract:
PROBLEM TO BE SOLVED: To provide a multiprocessor system to easily add an extension function peculiar to specified palliation software to a system while holding upward compatibility of the system.
SOLUTION: The system is provided with a means to assign a process to be executed in the system to a virtual processor, a means to generate status information whose contents are whether the process assigned to the virtual processor is the process peculiar to the specified application software or not, all purpose physical processors 1, 2, processors 3, 4 for the process peculiar to the specified application software and a means 5 to execute the virtual processor by switching the virtual processor to the all purpose processors 1, 2 and the processors 3, 4 for the proper process according to the contents of status information.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12)**公開特許公報**（Ａ）　(11)特許出願公開番号

特開平10－334055

(43)公開日　平成10年(1998)12月18日

(51)Int.Cl.[6]	識別記号	ＦＩ	
Ｇ０６Ｆ　15/16	３７０	Ｇ０６Ｆ　15/16	３７０Ｎ

審査請求　未請求　請求項の数１　ＯＬ　（全７頁）

(21)出願番号　特願平9－145383

(22)出願日　平成９年(1997)６月３日

(71)出願人　000002185
ソニー株式会社
東京都品川区北品川６丁目７番35号

(72)発明者　田村　征大
東京都品川区北品川６丁目７番35号　ソニー株式会社内

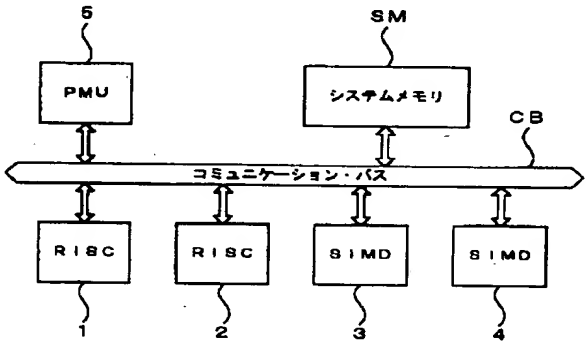
(74)代理人　弁理士　松隈　秀盛

(54)【発明の名称】　マルチプロセッサ・システム

(57)【要約】

【課題】　特定のアプリケーションソフトウェアに固有の拡張機能を、システムの上位互換性を保ったまま容易にシステムに追加できるようにしたマルチプロセッサ・システムを提供する。

【解決手段】　システムにおいて実行すべきプロセスを仮想的なプロセッサに割り当てる手段と、仮想的なプロセッサに割り当てられたプロセスが特定のアプリケーションソフトウェアに固有のプロセスであるか否かを内容とするステータス情報を作成する手段と、汎用物理プロセッサ１，２と、特定のアプリケーションソフトウェアに固有のプロセスのためのプロセッサ３，４と、ステータス情報の内容に応じて、仮想的なプロセッサを汎用プロセッサ１，２と固有のプロセスのためのプロセッサ３，４とに切り替えて実行させる手段５とを備えている。



【特許請求の範囲】
【請求項１】　システムにおいて実行すべきプロセスを仮想的なプロセッサに割り当てる手段と、
前記仮想的なプロセッサに割り当てられた前記プロセスが特定のアプリケーションソフトウェアに固有のプロセスであるか否かを内容とするステータス情報を作成する手段と、
汎用プロセッサと、
前記固有のプロセスのためのプロセッサと、
前記ステータス情報の内容に応じて、前記仮想的なプロセッサを前記汎用プロセッサと前記固有のプロセスのためのプロセッサとに切り替えて実行させる手段とを備えたことを特徴とするマルチプロセッサ・システム。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明は、複数のプロセッサを結合し、統一したＯＳ（オペレーティングシステム）のもとでハードウェア資源及びソフトウェア資源を共用した並列処理システムであるマルチプロセッサ・システムに関し、特に、特定のアプリケーションソフトウェアに固有の拡張機能を、システムの上位互換性を保ったまま容易にシステムに追加できるようにしたものに関する。
【０００２】
【従来の技術】従来、パーソナル・コンピュータ等で採用されているＳＭＰ（対称型マルチプロセッサ）システムをはじめとする各種マルチプロセッサ・システムにおいて、例えばマルチメディア処理のような特定のアプリケーションソフトウェアに固有の拡張機能を追加する場合には、システム内の汎用プロセッサに、当該処理のための特別な演算器等を付加するような大幅なアーキテクチャの変更を行っていた。
【０００３】
【発明が解決しようとする課題】このような汎用プロセッサ自体のアーキテクチャの大幅な変更を伴う拡張機能の追加を、システムの上位互換性を保ったまま行うことは、大変困難な作業であった。また、こうした拡張機能の使用頻度は通常はそれほど高くないので、その機能の追加のためにこうした困難な作業を行うことは、非効率的であった。
【０００４】本発明は上述の点に鑑みてなされたもので、特定のアプリケーションソフトウェアに固有の拡張機能を、システムの上位互換性を保ったまま容易にシステムに追加できるようにしたマルチプロセッサ・システムを提供しようとするものである。
【０００５】
【課題を解決するための手段】本発明に係るマルチプロセッサ・システムは、システムにおいて実行すべきプロセスを仮想的なプロセッサ（以下、仮想プロセッサと呼ぶ）に割り当てる手段と、仮想的なプロセッサに割り当てられた前記プロセスが特定のアプリケーションソフトウェアに固有のプロセスであるか否かを内容とするステータス情報を作成する手段と、汎用プロセッサと、特定のアプリケーションソフトウェアに固有のプロセスのためのプロセッサ（以下、システムに実装されているこれら汎用プロセッサ及び固有のプロセスのためのプロセッサを物理プロセッサとも呼ぶ）と、ステータス情報の内容に応じて、仮想的なプロセッサを汎用プロセッサと固有のプロセスのためのプロセッサとに切り替えて実行させる手段とを備えたことを特徴としている。
【０００６】このマルチプロセッサ・システムでは、実装されたプロセッサとは別の「仮想プロセッサ」の概念を導入しており、システムにおいて実行すべきプロセスを、この仮想プロセッサに割り当てる。従って、アプリケーションソフトウェアのレベルでは、仮想プロセッサの使用を前提としたプログラムの記述を行えば足り、システムに実装された物理プロセッサの使用を前提としたプログラムの記述を行う必要はなくなる。
【０００７】こうした仮想プロセッサの概念の導入によりアプリケーションソフトウェアのレベルでは物理プロセッサの使用を前提とする必要がなくなることを利用して、このマルチプロセッサ・システムでは、次のようにして、特定のアプリケーションソフトウェアに固有の拡張機能を、システムの上位互換性を保ったまま容易にシステムに追加できるようにしている。
【０００８】即ち、汎用プロセッサと、特定のアプリケーションソフトウェアに固有のプロセスのための特殊なプロセッサとの２通りの物理プロセッサを実装している。そして、仮想的なプロセッサに割り当てられた前記プロセスが特定のアプリケーションソフトウェアに固有のプロセスであるか否かを内容とするステータス情報を作成するようにしており、このステータス情報の内容に応じて、仮想的なプロセッサをこれらの汎用プロセッサと特殊なプロセッサとに切り替えて実行させるようにしている。
【０００９】これにより、仮想プロセッサに割り当てられたプロセスが特定のアプリケーションソフトウェアに固有のプロセスではない一般的なプロセスであるときには、仮想プロセッサが汎用プロセッサで実行され、他方、仮想プロセッサに割り当てられたプロセスが特定のアプリケーションソフトウェアに固有のプロセスであるときには、仮想プロセッサがその固有のプロセスのためのプロセッサで実行されるようになる。
【００１０】このように、このマルチプロセッサ・システムでは、仮想プロセッサの概念の導入を前提として、汎用プロセッサのアーキテクチャを全く変更しないことによりシステムの上位互換性を保ったまま、特定のアプリケーションソフトウェアに固有の拡張機能を容易にシステムに追加することができる。
【００１１】しかも、システムを構成する際に、一般的

なプロセスのための汎用プロセッサの数と、アプリケーションソフトウェアに固有のプロセスのためのプロセッサの数とを、システムで実行させるアプリケーションソフトウェアに応じて決定することにより、効率的なシステム構成を実現することができる。

【００１２】

【発明の実施の形態】以下、添付図面を参照して本発明の実施例を詳細に説明する。

《１．システムの構成例》図１は、本発明に係るマルチプロセッサ・システムの構成の一例を概念的に示す。このシステムには、ＯＳやアプリケーションソフトウェア等を実行するためのプロセッサエレメントとして、２基の汎用のＲＩＳＣプロセッサ１，２と、特定用途向けの（特定のアプリケーションソフトウェアに固有のプロセスのための）２基のＳＩＭＤ（単一命令複数データ流方式）プロセッサ３，４とが設けられている。また、これらのプロセッサ１～４の他に、プロセッサ・マネージメント・ユニット（ＰＭＵ）５が設けられている。ＰＭＵ５は、仮想プロセッサを管理するための一種のコプロセッサである。

【００１３】各プロセッサ１～４とＰＭＵ５とは、コミュニケーション・バスＣＢを介して接続されている。コミュニケーション・バスＣＢは、接続されている任意のエレメント間で多ビットの情報のやり取りが可能な双方向バスである。システムの主記憶装置であるシステムメモリＳＭも、このコミュニケーション・バスＣＢを介して各プロセッサ１～４に接続されることにより、各プロセッサ１～４に共有されている。

【００１４】〈１ａ．ＰＭＵとＰＥＴＬＢ〉このマルチプロセッサ・システムでは、仮想プロセッサと、物理プロセッサである各プロセッサ１～４とに、それぞれ固有のＩＤ（識別ラベル）が与えられている。仮想プロセッサのＩＤを「ＶＥＩＤ」，物理プロセッサのうちＲＩＳＣプロセッサ１，２のＩＤを「ＰＥＩＤ」，物理プロセッサのうちＳＩＭＤプロセッサ３，４のＩＤを「ＳＰＥＩＤ（Ｓｐｅｃｉａｌ　ＰＥＩＤ）」と呼ぶ。

【００１５】ＶＥＩＤは、後出の〈４ｂ．仮想プロセッサの生成〉で述べるように、システムの起動後新たに仮想プロセッサを生成するときに、ＯＳによってその仮想プロセッサに与えられたものである。ＰＥＩＤとＳＰＥＩＤは、システムの構成に応じてハードウェアによって与えられたものであり、変更することはできない。

【００１６】ＰＭＵ５（図１）内には、仮想プロセッサのＶＥＩＤと、その仮想プロセッサのステータス情報である「実行レベル」と、この仮想プロセッサへの割り当ての候補として定義される（この定義も、後出の〈４ｂ．仮想プロセッサの生成〉で述べるように、システムの起動後新たに仮想プロセッサを生成するときに、ＯＳによって行われる）物理プロセッサ１～４のＩＤ（ＰＥＩＤ及びＳＰＥＩＤ）とを関連付けて記憶させることのできる記憶領域（テーブル）が用意されている。このテーブルを、プロセッサエレメントテーブル（ＰＥＴＬＢ）と呼ぶ。

【００１７】実行レベルは、通常のマイクロプロセッサにおける特権レベルあるいは保護モードと同義のものか、あるいはそれを拡張させたものである。仮想プロセッサに割り当てられたプロセスが特定のアプリケーションソフトウェアに固有のプロセスではない一般的なプロセスである場合には、実行レベルはユーザー・レベルあるいはスーパーバイザ・レベルとなっている。他方、仮想プロセッサに割り当てられたプロセスが特定のアプリケーションソフトウェアに固有のプロセスである場合には、実行レベルはアプリケーション・レベルになっている。仮想プロセッサへのこの実行レベルの付与も、後出の〈４ｂ．仮想プロセッサの生成〉で述べるように、システムの起動後新たに仮想プロセッサを生成するときに、ＯＳによって行われる。

【００１８】図２は、プロセッサエレメントテーブル（ＰＥＴＬＢ）の一例を概念的に示す。ＰＥＴＬＢは同図Ａのように複数のエントリを持っており、各エントリ（但し図の最上部のシステムエントリＳＥを除く）は同図Ｂのようにいくつかのフィールドに分かれている。

【００１９】これらのフィールドのうち、Ｖａｌｉｄフィールド６は、そのエントリに有効なデータが存在することを示すためのフィールドであり、Ｌｏｃｋフィールド７は、そのエントリの内容を変更することが禁止されていることを示すためのフィールドである。

【００２０】同一エントリのＬｅｖｅｌフィールド８，ＶＥＩＤフィールド９，ＰＥＩＤフィールド１０，ＳＰＥＩＤフィールド１１には、仮想プロセッサの実行レベル，その仮想プロセッサのＶＥＩＤ，その仮想プロセッサへの割り当ての候補として定義されたＲＩＳＣプロセッサ１または２のＰＥＩＤ，その仮想プロセッサへの割り当ての候補として定義されたＳＩＭＤプロセッサ３または４のＳＰＥＩＤが、それぞれ格納される。

【００２１】各エントリに記憶される内容は、ＯＳによってシステムメモリＳＭ（図１）内で管理されている「プロセッサ・エレメント管理テーブル」（仮想プロセッサと物理プロセッサとの対応を定義するいわば仮想プロセッサの名簿）の一部分のコピーである。ＰＥＴＬＢは一種の連想記憶装置になっており、ＶＥＩＤを与えたとき、そのＶＥＩＤの仮想プロセッサに割り当てられるＲＩＳＣプロセッサ１または２のＰＥＩＤやＳＩＭＤプロセッサ３または４のＳＰＥＩＤを引き出せるようになっている。

【００２２】尚、システムエントリＳＥは、後述のシステム・プロセッサに対する物理プロセッサＰＥ２の割り当て情報を管理するための特別なエントリである。

【００２３】ＰＥＴＬＢのこうした構造は、一般的なマイクロプロセッサの仮想記憶方式に置いて用いられるテ

ーブルと似ている。

【００２４】〈１ｂ．システム・プロセッサ〉「システム・プロセッサ」は、仮想プロセッサの中の一つであり、システム全体に関わるエラーやイベントを処理するための特別な仮想プロセッサである。システム外部からの割り込みや、仮想プロセッサの管理に関するエラー等は、このシステム・プロセッサで発生したエラーとして処理される。

【００２５】システム・プロセッサ以外の一般の仮想プロセッサについてのエントリの初期化は、通常はソフトウェアによってシステムメモリＳＭ内のプロセッサ・エレメント管理テーブルで行わなければならないが、システム・プロセッサだけは、後出の〈３ａ．システムの初期化〉で述べるように、システムのリセット時にハードウェアによってＰＥＴＬＢでシステムエントリＳＥが初期化される。

【００２６】また、ＰＥＴＬＢのシステムエントリＳＥ以外のエントリは、後出の《２．仮想プロセッサへのアクセス》で述べるようにプロセッサ・エレメント管理テーブル中の他のエントリと入れ替えられることがあるが、システムエントリＳＥだけは、この入れ替えからは保護された位置にある。換言すれば、システム・プロセッサだけには常に物理プロセッサが割り当てられており、システムエントリＳＥからその物理プロセッサを検索することができる。但し、システム・プロセッサにどの物理プロセッサを割り当てるは、システムエントリＳＥ自体を書き換えることによっていつでも変更することが可能である。

【００２７】《２．仮想プロセッサへのアクセス》アプリケーションソフトウェアがシステム中のプロセッサにアクセスする場合、（例えば現在プロセスを処理中のプロセッサとは別のプロセッサにプロセスを投入したり、現在プロセスを処理中のプロセッサとは別のプロセッサと通信を行ったりするような場合等）には、ＶＥＩＤを用いて仮想プロセッサに対してアクセスするようにする。換言すれば、アプリケーションソフトウェアでは、ＶＥＩＤを用いて、仮想プロセッサの使用を前提としたプログラムの記述を行うようにする。

【００２８】現在プロセスを処理中の物理プロセッサ（即ちプロセッサ１～４（図１）のうち現在実行中の仮想プロセッサに割り当てられた物理プロセッサ）の中で、このような別の仮想プロセッサに対するアクセスが発生すると、それが仮想プロセッサの呼び出しイベントとしてコミュニケーション・バスＣＢ（図１）上に通知される。このとき、アクセスされた仮想プロセッサのＶＥＩＤも同時にバスＣＢ上に乗せられる。

【００２９】各物理プロセッサ１～４は、それぞれ実行中の仮想プロセッサのＶＥＩＤを記憶するための「ＶＥＩＤレジスタ」を持っており、このレジスタを参照することにより、現在どの仮想プロセッサに割り当てられているかを判別することができる。物理プロセッサは、バスＣＢ上に乗せられたＶＥＩＤと現在割り当てられている仮想プロセッサのＶＥＩＤとが一致した場合（即ち自分が呼び出された仮想プロセッサである場合）、アクセスに対して直接応答する。

【００３０】一方、ＰＭＵ５（図１）は、ＰＥＴＬＢ（図２）の各エントリから、バスＣＢを介して送られたＶＥＩＤを格納しているＶＥＩＤフィールド９を検索する。そして、そのＶＥＩＤフィールド９と同じエントリ中から、当該仮想プロセッサに割り当てられる物理プロセッサのＩＤ（ＰＥＩＤまたはＳＰＥＩＤのいずれか）を得る（この部分の処理については、拡張機能と関連する処理なので、後出の《３．拡張機能の使用方法》で再度詳述する）。そして、いずれの物理プロセッサ１～４もアクセスに対して直接応答しなかったときには、得られたＩＤを持つ物理プロセッサに対して、仮想プロセッサの入れ替えのための割り込み要求を発生する。

【００３１】このＩＤを持つ物理プロセッサでは、この割り込みにより、ＶＥＩＤレジスタ内のＶＥＩＤの書き換えと、例外ハンドラ（例外処理を記述したプログラム）によるプロセスの入れ替えとが行われる。

【００３２】尚、ＰＭＵ５がＰＥＴＬＢの各エントリを検索した際に、バスＣＢを介して送られたＶＥＩＤを格納しているＶＥＩＤフィールド９がみつからなかった場合には、前述のシステム・プロセッサに例外が発生する。この場合には、ＯＳが例外ハンドラによってＰＥＴＬＢのエントリの入れ替えを行い、その後改めて仮想プロセッサの呼び出しが行われる。

【００３３】《３．拡張機能の使用方法》《２．仮想プロセッサへのアクセス》で述べたように、ＰＭＵ５（図１）は、ＰＥＴＬＢ（図２）の各エントリから、バスＣＢを介して送られたＶＥＩＤを格納しているＶＥＩＤフィールド９を検索するが、そのＶＥＩＤフィールド９と同じエントリ中のＬｅｖｅｌフィールド８の実行レベルがユーザー・レベルあるいはスーパーバイザ・レベルである場合（即ち、呼び出された仮想プロセッサに割り当てられたプロセスが特定のアプリケーションソフトウェアに固有のプロセスではない一般的なプロセスである場合）には、ＰＭＵ５は、そのエントリ中のＰＥＩＤフィールド１０からＰＥＩＤを得る。これにより、当該仮想プロセッサへの割り当ての候補として定義された物理プロセッサのうち、ＲＩＳＣプロセッサ１または２が実際に当該仮想プロセッサに割り当てられる。

【００３４】これに対し、そのエントリ中のＬｅｖｅｌフィールド８の実行レベルがアプリケーション・レベルである場合（即ち、呼び出された仮想プロセッサに割り当てられたプロセスが特定のアプリケーションソフトウェアに固有のプロセスである場合）には、ＰＭＵ５は、そのエントリ中のＳＰＥＩＤフィールド１１からＳＰＥＩＤを得る。これにより、当該仮想プロセッサへの割り

当ての候補として定義された物理プロセッサのうち、ＳＩＭＤプロセッサ３または４が実際に当該仮想プロセッサに割り当てられる。

【００３５】そして、《２．仮想プロセッサへのアクセス》で述べたように、こうして割り当てられた物理プロセッサで、ＶＥＩＤレジスタ内のＶＥＩＤの書き換えとプロセスの入れ替えとが行われる。

【００３６】次に、現在実行中の仮想プロセッサに割り当てられたプロセスが特定のアプリケーションソフトウェアに固有のプロセスではない一般的なプロセスと特定のアプリケーションソフトウェアに固有のプロセスとの間で変化すると、当該仮想プロセッサを実行中の物理プロセッサが特定の命令を実行することにより、ユーザー・レベルあるいはスーパーバイザ・レベルとアプリケーション・レベルとの間での実行レベルの遷移が起こる。この実行レベルの遷移は、仮想プロセッサの呼び出しイベントとしてコミュニケーション・バスＣＢ（図１）上に通知される。ＰＭＵ５（図１）は、この呼び出しイベントで呼び出された仮想プロセッサについてのＰＥＴＬＢ上のエントリ中のＬｅｖｅｌフィールド８をこの遷移した実行レベルに更新した後、当該仮想プロセッサへの物理プロセッサの割り当てを行う。

【００３７】この実行レベルの遷移時の物理プロセッサの割り当てには、次の３通りのパターンがある。

〈３ａ．別の物理プロセッサへの割り当て（再割り当て）を伴わない実行レベルの遷移が起こった場合〉この場合には、それまで当該仮想プロセッサの実行が割り当てられていた物理プロセッサに、引き続き当該仮想プロセッサの実行を継続させる。

【００３８】〈３ｂ．再割り当てを伴う実行レベルの遷移が起こり、且つ、新たに割り当てられた物理プロセッサがアイドル状態である場合〉この場合には、ＰＭＵ５が、新たに割り当てられた物理プロセッサ（ユーザー・レベルあるいはスーパーバイザ・レベルに遷移した場合にはＲＩＳＣプロセッサ１または２、アプリケーション・レベルに遷移した場合にはＳＩＭＤプロセッサ３または４）に対して、仮想プロセッサの入れ替えのための割り込み要求を発生することにより、その物理プロセッサに当該仮想プロセッサを実行させる。それまで当該仮想プロセッサの実行が割り当てられていた物理プロセッサは、アイドル状態に移行する。

【００３９】〈３ｃ．再割り当てを伴う実行レベルの遷移が起こり、且つ、新たに割り当てられた物理プロセッサが他の仮想プロセッサを実行中である場合〉この場合には、それまで当該仮想プロセッサの実行が割り当てられていた物理プロセッサはアイドル状態に移行するが、新たに割り当てられた物理プロセッサでは、後出の〈４ｃ．仮想プロセッサの実行〉で述べるようにタイマ割り込みのような外部割り込みによってシステム全体のタスク切り替えが行われるまで、当該仮想プロセッサの実行が開始されない。

【００４０】《４．ソフトウェアとの連携による仮想プロセッサ管理の例》

〈４ａ．システムの初期化〉システムがリセットされると、各物理プロセッサ１～４は初期化されて待機状態になる。ＰＥＴＬＢ（図２）は、システムエントリＳＥが初期化され、それ以外のエントリが全て無効化される。

【００４１】すべての初期化処理が終了すると、ＰＭＵ５（図１）がシステム・プロセッサを呼び出す。この呼び出しは、《２．仮想プロセッサへのアクセス》で述べた物理プロセッサに対する割り込み要求によって行われる。

【００４２】起動したシステム・プロセッサでは、ＯＳのカーネル（核）の実行が開始され、システムメモリＳＭ（図１）内のプロセッサ・エレメント管理テーブルの初期化が行われる。

【００４３】〈４ｂ．仮想プロセッサの生成〉その後、ＯＳのその他の部分やアプリケーションソフトウェアが起動されて、それぞれのプロセスが新しい仮想プロセッサに割り当てられる。この処理は、プロセスが、ＯＳの用意したプログラムである「仮想プロセッサ生成タスク」をシステム・コールで呼び出すことによって行われる。

【００４４】呼び出された仮想プロセッサ生成タスクは、プロセスを割り当てられた仮想プロセッサにＶＥＩＤを与え、その仮想プロセッサに実行レベルを付与し（割り当てられたプロセスが特定のアプリケーションソフトウェアに固有のプロセスではない一般的なプロセスであるときはユーザー・レベルあるいはスーパーバイザ・レベルを付加し、割り当てられたプロセスが特定のアプリケーションソフトウェアに固有のプロセスであるときはアプリケーション・レベルを付与し）、その仮想プロセッサへの割り当ての候補となるＲＩＳＣプロセッサ１または２とＳＩＭＤプロセッサ３または４とを定義する。そして、そのＶＥＩＤと実行レベルとＰＥＩＤとＳＰＥＩＤと当該プロセスの実行に必要な情報とを、システムメモリＳＭ（図１）内のプロセッサ・エレメント管理テーブルに記憶させる。このようにして、仮想プロセッサが生成される。

【００４５】尚、新しく生成する仮想プロセッサへの割り当ての候補としてＲＩＳＣプロセッサ１，２のうちのいずれを割り当て、ＳＩＭＤプロセッサ３，４のうちのいずれを割り当てるかは、ＯＳが各物理プロセッサ１～４の使用状況等をもとにして決定する。ＶＥＩＤで表現可能な範囲内で、物理プロセッサの総数である４よりも多い数の仮想プロセッサを生成することも可能であり、その場合には複数の仮想プロセッサに同一の物理プロセッサを共有させるようにする。

【００４６】〈４ｃ．仮想プロセッサの実行〉新しく生成された仮想プロセッサを実際に物理プロセッサ１～４

上で実行させる処理は、仮想プロセッサの生成処理とは独立かつ並行的に行われる。

【００４７】そのために、システムには、タイマ割り込みのような外部割り込みが定期的にかかるようになっている。この仕組みは、単一のプロセッサに時分割によりマルチタスクを行わせるためのものと同じである。

【００４８】タイマ割り込みはシステム全体に関係するイベントなので、それが発生すると、ＰＭＵ５が物理プロセッサに対する割り込み要求によって自動的にシステム・プロセッサを呼び出す。この割り込み処理を記述したプログラムである割り込みハンドラから、ＯＳの用意したプログラムである「仮想プロセッサ管理タスク」が呼び出されるようになっている。

【００４９】呼び出された仮想プロセッサ管理タスクは、システムメモリＳＭ内のプロセッサ・エレメント管理テーブルから、次のタイマ割り込みまでの一定期間物理プロセッサを割り当てて実行する仮想プロセッサを、各プロセス間の優先順位等をもとに決定する。こうして決定された仮想プロセッサが、《２．仮想プロセッサへのアクセス》で述べたようアプリケーションソフトウェアによってアクセスされることにより、その仮想プロセッサの実行が開始される。

【００５０】以上のように、このマルチプロセッサ・システムによれば、仮想プロセッサの導入を前提として、汎用の物理プロセッサであるＲＩＳＣプロセッサ１，２のアーキテクチャを全く変更しないことによりシステムの上位互換性を保ったまま、特定のアプリケーションソフトウェアに固有の拡張機能を容易にシステムに追加することができる。

【００５１】尚、以上の実施例では汎用の物理プロセッサであるＲＩＳＣプロセッサと特定用途向けのＳＩＭＤプロセッサとをそれぞれ２基ずつ設けているが、これらのプロセッサをそれぞれ適宜の数だけ設けるようにしてよいことはもちろんである。特に、システムを構成する際に、システムで実行させるアプリケーションソフトウェアに応じてこれらのプロセッサの数を適宜決定することにより、効率的なシステム構成を実現することができる。

【００５２】また、本発明が前提とするマルチプロセッサ・システムは、以上の実施例に示したものに限らず、仮想プロセッサの概念を導入したシステムであれば適宜のものであってよい。

【００５３】また、以上の実施例では、主記憶装置（システムメモリＳＭ）を単一のバス（コミュニケーション・バスＣＢ）を介して各プロセッサ（物理プロセッサ１～４）に共有させる単一バス方式の共有メモリ形（密結合形）マルチプロセッサに本発明を適用しているが、その他の方式（例えば多重バス方式，階層バス方式，マルチポートメモリ方式等）の共有メモリ形マルチプロセッサや、あるいはメッセージ交換形（疎結合形）マルチプロセッサに本発明を適用するようにしてもよい。

【００５４】また、本発明は、以上の実施例に限らず、本発明の要旨を逸脱することなく、その他様々の構成をとりうることはもちろんである。

【００５５】

【発明の効果】以上のように、本発明に係るマルチプロセッサ・システムによれば、汎用プロセッサのアーキテクチャを全く変更しないことによりシステムの上位互換性を保ったまま、特定のアプリケーションソフトウェアに固有の拡張機能を容易にシステムに追加することができる。

【００５６】また、システムを構成する際に、一般的なプロセスのための汎用プロセッサの数と、アプリケーションソフトウェアに固有のプロセスのためのプロセッサの数とを、システムで実行させるアプリケーションソフトウェアに応じて決定することにより、効率的なシステム構成を実現することができる。

【図面の簡単な説明】

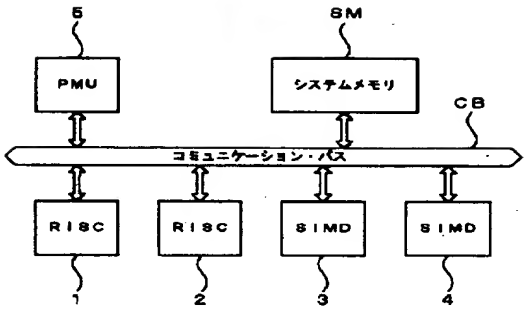
【図１】本発明に係るマルチプロセッサ・システムの構成の一例を示す図である。

【図２】プロセッサエレメントテーブルの一例を示す図である。

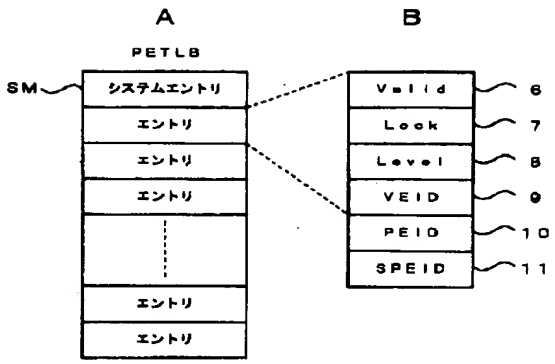
【符号の説明】

１，２　ＲＩＳＣプロセッサ、　３，４　ＳＩＭＤプロセッサ、　５　プロセッサ・マネージメント・ユニット、　ＣＢ　コミュニケーション・バス、　ＳＭシステムメモリ、　６　Ｖａｌｉｄフィールド、　７　Ｌｏｃｋフィールド、８　Ｌｅｖｅｌフィールド、　９　ＶＥＩＤフィールド、　１０　ＰＥＩＤフィールド、　１１　ＳＰＥＩＤフィールド

【図1】


5
SM
PMU
システムメモリ
CB
コミュニケーション・バス
RISC
RISC
SIMD
SIMD
1
2
3
4

【図2】


A
B
PETLB
SM
システムエントリ
エントリ
エントリ
エントリ
エントリ
エントリ
Valid
Lock
Level
VEID
PEID
SPEID
6
7
8
9
10
11